Autonomous Receiver Configuration

Jon Hamkins
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Dr., M/S 238-420
Pasadena, CA 91109–8099
Email: jon.hamkins@jpl.nasa.gov

***Abstract*— Space missions are being developed with increasing levels of autonomy, as a means to increase their science return, fault-tolerance, and longevity. In this paper, we present an architecture of an autonomous radio that can correctly receive a signal without precise a priori knowledge of its data rate, carrier frequency, modulation type, modulation index, and other defining characteristics. This is accomplished by estimating and classifying these parameters directly from the observed signal, enabling the receiver to configure itself. We describe the function of each estimator/classifier module and how they can interact to ultimately recover the transmitted message.**

I. Introduction

NASA's Mars Exploration Program Advanced Technologies is funding an ambitious project to develop and demonstrate an autonomous radio capable of receiving a signal without precise a priori knowledge of its carrier frequency, modulation type, modulation index, data rate, and other characteristics [1]. This contrasts with a conventional radio, which must be explicitly configured with these parameters. As a first step in this direction, a suite of modules has been developed to autonomously recognize various signal attributes, including the angle of arrival, data rate, symbol timing, carrier frequency and phase, modulation index, modulation type, signal-to-noise ratio (SNR), code type, and decoded message bits. This paper is an overview of the architecture of the autonomous radio receiver, describing what each module does and how they interact to produce the desired effect.

The primary application of this technology is in relaying communication signals from multiple deep space assets. For example, we might desire for two or more rovers on a distant planet to relay data through an orbiter, as the two Mars Exploration Rovers have done via Mars Global Surveyor and Mars Odyssey [2]. Multiple landed assets communicating through relays will continue to be an important part of NASA's exploration plans throughout the next two decades [3], [4]. Over a period of years, we may expect NASA and other space agencies to launch a set of diverse orbiters and landers[1], and because technology continues to emerge, it is unlikely that they will all use the same data rates, protocols, error-correcting codes, and modulation types.

The advantage of an autonomous radio in this emerging scenario is that it can communicate to each asset that comes into view, automatically, without having to be reconfigured from Earth for each pass to account for differences in the signal characteristics. The radio would receive whatever each landed asset sent. Since an orbital period may be a few hours, an orbiter may come within view of various landed assets several times a day, and the automatic reconfiguration would be this frequent. By comparison, manual radio reconfiguration would be a daunting task. The reduced burden of manual receiver configuration also makes the technology attractive for ground stations.

In addition to easing the scheduling and configuration burden, an autonomous radio will also gracefully handle unpredictable or anomalous events. For example, during entry, descent, and landing (EDL), a spacecraft can undergo large Doppler swings caused by rocket firings, parachute openings, backshell ejection, and a bouncing landing on the surface. Even when all scheduled events occur successfully, there may be Doppler uncertainty due to unpredictable properties of the atmosphere. Ideally, the communication link should operate whether or not each of the EDL events is successful, but the uncertainties involved typically lead to liberal link margins— for example, the Mars Exploration Rovers observed link margins that sometimes exceeded 10 dB. An autonomous radio could substantially reduce this design margin because it would handle any Doppler swing nearly optimally.

Such flexibile technology is not yet available on NASA's currently flying missions. In perhaps the most glaring example of this, NASA engineers discovered in 2000 that a receiver aboard Cassini, launched in 1997, would fail during the Huygens probe descent onto Titan because it did not properly account for the Doppler profile of the probe [5]. Increasing the loop bandwidth of the synchronization loops would have easily fixed the problem, but unfortunately, these loop bandwidths were hard wired to set values on the spacecraft. With superior engineering and enormous dedication, NASA/ESA was able to save the mission by slightly altering the original trajectory, but this solution required forming a large and expensive international recovery team to find the appropriate recommendations on how to overcome the radio's limitations.

II. Preliminaries

A. Mathematical signal model

This paper considers a single channel amplitude and phase modulated signal with or without a residual carrier. In complex

The research described in this publication was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

[1]See http://www.jpl.nasa.gov/missions/future_missions.cfm

baseband, the transmitted signal has the form

$$\tilde{s}(t) = \sum_{n=-\infty}^{\infty} A_n p(t-nT) e^{j[\theta_n + \theta_c(t)]} + \sqrt{2P_c} e^{j\theta_c(t)} \quad (1)$$

where A_n is the amplitude of the n^{th} symbol; θ_n is the phase modulation for the n^{th} M-PSK symbol; $\theta_c(t)$ is the carrier phase; $p(t)$ is a pulse shape satisfying $T^{-1}\int_0^T p^2(t)dt = 1$; T is the symbol duration; and P_c is the residual carrier power of the passband signal $s(t) = \text{Re}\{\tilde{s}(t)e^{2\pi j f_c t}\}$, where f_c is the carrier frequency.

When the amplitude is not modulated, as is the case with phase-shift keying (PSK), we may write $A_n = \sqrt{2P_d}$, a constant equal to the data modulation passband signal power. In the special case in which $\theta_n = c_n\pi/2$ with $c_n \in \{-1, 1\}$, and $p(t)$ a digital pulse shape taking on values ± 1, i.e., a Binary PSK (BPSK) signal, (1) becomes

$$\tilde{s}(t) = \sum_{n=-\infty}^{\infty} \sqrt{2P_t} p(t-nT) e^{j[\beta c_n + \theta_c(t)]}, \quad (2)$$

where $P_t = P_c + P_d$ is the total passband signal power and $\beta = \tan^{-1}\sqrt{P_d/P_c}$ is the modulation index.

At the receiver, the timing and carrier phase are initially unknown, and noise is present. If we also assume that the carrier frequency f_c is imperfectly estimated as $\hat{f}_c$ at the front-end of the receiver, then a residual frequency component $f_r = f_c - \hat{f}_c$ will remain after conversion to baseband, and the resulting signal will have the form

$$\tilde{r}(t) = \tilde{s}(t) + \tilde{n}(t) \quad (3)$$

$$= \sum_{n=-\infty}^{\infty} A_n p(t-(n-\varepsilon)T) e^{j[2\pi f_r t + \theta_n + \theta_c(t)]} \quad (4)$$

$$+ \sqrt{2P_c} e^{j[2\pi f_r t + j\theta_c(t)]} + \tilde{n}(t), \quad (5)$$

where ε is the unknown symbol timing and $\tilde{n}(t) = \sqrt{2}[n_c(t) + jn_s(t)]$, where $n_c(t)$ and $n_s(t)$ are each an additive white Gaussian noise (AWGN) process with two-sided power-spectral density $N_0/2$ W/Hz. A priori, ε is uniformly distributed on $[0, T)$ and θ_c is uniformly distributed[2] on $[0, 2\pi)$.

B. Anatomy of the received signal.

Fig. 1 graphically indicates the dependence of the received signal on several factors. We group the signal dependence graph into three primary components: the forward error-correcting (FEC) code encoder, the modulator/amplifier, and the channel. Each of these is affected by several subfactors, including the ones shown in Fig. 1 as well as others which we call out in italics in the more detailed discussion below.

[2]For our purposes here, we assume that the carrier phase $\theta_c(t)$ is slowly varying with respect to the date rate; thus, we drop the dependence on time in the notation.

1) FEC code: The FEC code can be one of several *code types*. The code types standardized by the Consultative Committee for Space Data System (CCSDS) for deep space [6] or in situ [7] communications include Reed-Solomon (RS) codes, convolutional codes, turbo codes, Bose-Chaudhuri-Hocquenghem (BCH) codes, and cyclic redudancy check codes (CRC). Work is also progressing rapidly both academically and in various standards (Digital Video Broadcast/Satellite, IEEE 802.{11n,15.3a,16e}, and CCSDS deep space and in situ) on low-density parity-check codes and progressive-parity type codes such as tornado and raptor codes.

Associated with each FEC code is its *code rate*, which is the fraction of symbols carrying information, and its *code length*, which indicates the number of symbols in each codeword. For some code types, these parameters alone are nearly enough to completely identify the code. For example, the best-performing convolutional codes for a given rate and constraint length are tabulated in textbooks (e.g., [8]), and applications using convolutional codes nearly always use codes from these tables. CRC codes of a given length also typically use standard generator polynomials [7]. RS codes are specified by their blocklength, rate, field generator polynomial, and code generator polynomials. The latter two can be one of several possibilities, but in practice space communication systems have primarily used the one that is specified in the CCSDS standard [6].

The techniques of *puncturing*, *shortening*, and *expurgating* are commonly used to modify a code. Puncturing raises the code rate, and it is the standard technique to obtain the CCSDS turbo codes of rate 1/4, 1/3, and 1/2 from the rate 1/6 mother code [6]. Shortening is often used with the standard RS(255,223) code— the full-length code with interleaving depth 5 has input-length $223\times 8\times 5 = 8920$, but missions often shorten this to 8800, a multiple of 32, which is a convenient quantity for spacecraft processors to handle. The (63,56) BCH code used for uplink commanding [9] is an expurgated (63,57) Hamming code, with odd-weight codewords removed.

In addition to the substantive factors mentioned above, there are a number of superficial factors that determine the FEC encoder output. The precise *bit ordering*, use of *trellis termination*, and *placement of frame headers, synchronization bits, and filler bits* are examples of these factors.

2) Modulator and amplifier: The modulator uses the coded binary sequence from the output of the FEC encoder to modulate a carrier signal. This process depends on several factors. The *modulation type* identifies the signal constellation from which the transmitted symbols are chosen. BPSK, quadrature PSK (QPSK), quadrature amplitude modulation (QAM), and Gaussian-filtered minimum-shift keying (GMSK) are commonly used modulation types [10]. In the case of GMSK, and other filtered modulation types, the bandwidth time (BT) product is also needed to fully specify the modulation.

The assignment of FEC encoded bits to symbols is defined by a *mapping*, which may be a static mapping such as a natural ordering, Gray code, or anti-Gray code, which maps each block of bit(s) to a symbol; or, the mapping may be

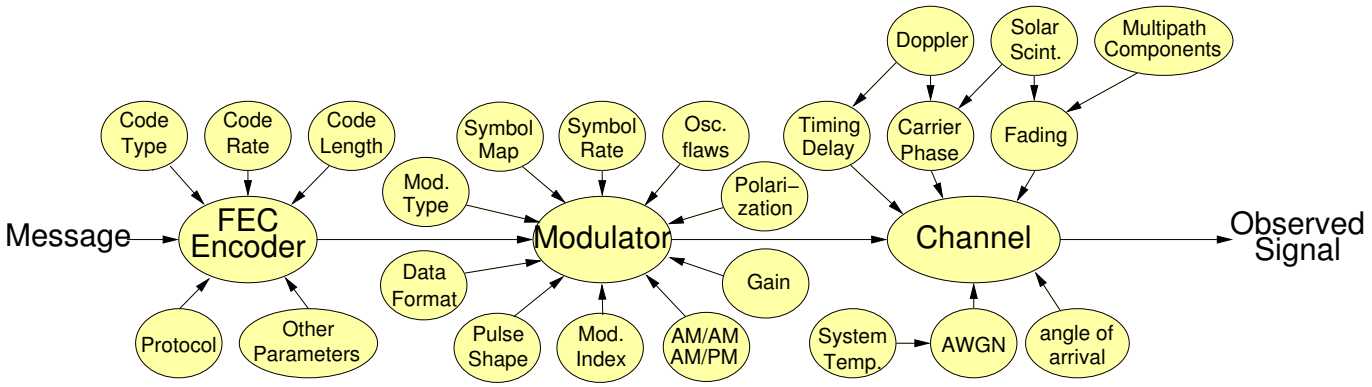

Fig. 1. Signal dependency graph.

dynamically controlled through a state machine, as it is with trellis coded modulation [11].

The *symbol rate*, or baud, defines the number of discrete signal constellation elements transmitted per second. Within each symbol epoch, a *pulse shape* (rectangular, raised-cosine, etc.) is applied. With BPSK signaling, the *data format* may be NRZ (non-return to zero) or Manchester encoded. The *modulation index* determines the fraction of total power that is allocated to an unmodulated carrier signal.

The carrier signal to be modulated is generated by an imperfect oscillator, whose quality can be measured by it's spectrum, or by distilling its spectrum to a single quantity such as *Allan deviation*, *phase noise* at a given offset, or *drift rate*. Ultrastable oscillators can achieve a phase noise of -100 dBc/Hz at a 1 Hz offset [12], although not all missions have the mass budget to carry one on board the spacecraft.

There are several signal-dependent factors in the amplifier, as well. Nominally, the amplifier output is larger than the input by the *gain* of the amplifier. However, depending on the input, distortion may affect the amplitude or phase. *AM/AM (amplitude modulation to amplitude modulation) distortion* occurs when the amplitude of the amplifier output is not proportional to the amplifier input. *AM-to-PM (AM to phase modulation) conversion* occurs when variations in the input amplitude result in unwanted phase modulation. Additionally, the *group delay* is the rate of change of the total phase shift with respect to angular frequency, and the *polarization* (right- or left-handed, circular or elliptical) describes the time-varying direction and amplitude of the electric field vector propagated from the transmitter.

3) Channel: Typically, deep space communications channels are quite benign, with *AWGN* being the dominating impairment. If *fading* is present, it may be due to *multipath interference* or *solar scintillations* caused by a small sun-earth-probe angle. *Doppler* affects carrier and timing parameters. The *angle of arrival*, *symbol timing*, and *carrier phase* are also modeled in the channel component of the dependency graph shown in Fig. 1.

III. Radio Receiver Architectures

A. A conventional radio receiver

A functional block diagram of a radio receiver and decoder is shown in Fig. 2. Factors that are known a priori in a conventional radio are shown in ellipses, while the tasks it performs are shown in rectangles. A conventional radio receiver has complete a priori knowledge of the signal-dependent factors relating to the FEC encoder and modulator/amplifier components shown in Fig. 1. Only the channel-related factors are not completely known— although, even those may be partially known through the use of predicts.

Knowledge of the transmitted signal parameters greatly simplifies the design and implementation of the receiver. For example, if a residual carrier is present, then the carrier phase tracking loop may be a simple phase-locked loop (PLL), and no Costas loop need be implemented. Or, if the modulation type is known to be BPSK, then the receiver need not include any processing of the quadrature component of the signal. Every rectangular block in Fig. 2 is similarly simplified by knowing the basic properties of the transmitted signal.

On the other hand, a conventional radio does not usually have much capability to receive signals types different from the single signal-type for which it was primarily designed, or, when it does have such capability, it requires specific pre-configuration according to a predetermined schedule. For example, if the radio can receive both suppressed carrier and residual carrier BPSK signals, it would typically carry both PLL and a Costas loops, or a hybrid loop that incorporates both components and would have to be preconfigured to use the appropriate loop (or to set the relative gains in the hybrid loop) based on knowledge of when each type of signal will be arriving. This is the approach taken by the highly capable advanced receiver design [13] that eventually became the Deep Space Network Block V Receiver.

B. Electra

Electra is NASA's first highly capable software defined radio [14]. Unlike other massively flexible radios, the Electra radio is an elegant, compact design based around a reprogrammable rad-tolerant Field Programmable Gate Array (FPGA). The

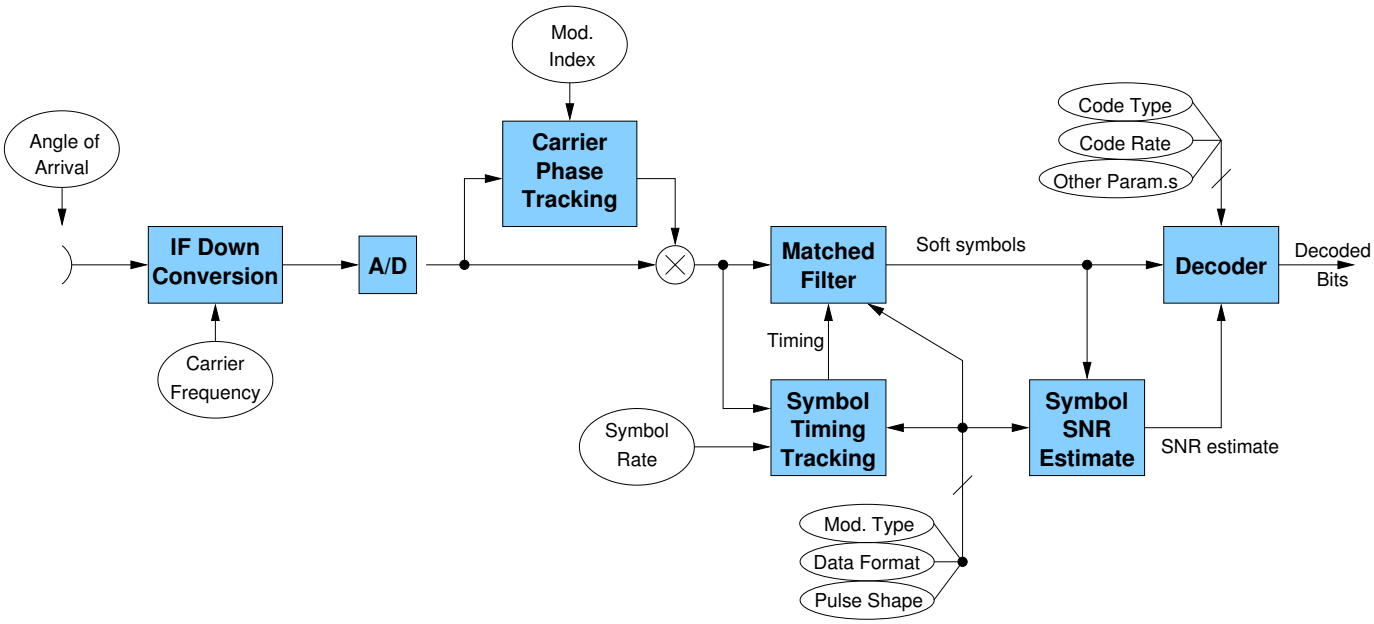

Fig. 2. Functional block diagram of a radio receiver. A conventional radio assumes a priori knowledge of all parameters shown in the ellipses, while an autonomous radio estimates them based on the incoming signal.

FPGA performs all the baseband processing for reception including carrier tracking, timing recovery, and demodulation. It also includes all the baseband processing necessary for transmission.

Unlike the Block V Receiver, the massive capability of this radio is not achieved through multiple simultaneous implementations of tracking loops and demodulators for all the various signal types it might encounter in its lifetime. Rather, the radio is simply redefined in the same small footprint by reprogramming the baseband processor module. This compact, flexible design makes it ideally suited for in situ radio radios, and in fact, it is now the NASA standard in situ radio and will fly on the Mars Reconnaisance Orbiter, Mars Telecommunications Orbiter, and Mars Science Laboratory missions, among others.

C. An autonomous radio

The fundamental difference between a conventional radio or even a software-defined radio such as Electra, and a truly autonomous radio, is that an autonomous radio has the ability to recognize features of an incoming signal and to respond intelligently, without explicit pre-configuration or reprogramming to define the functions of the radio.

In an autonomous radio, the parameters shown in ellipses in the functional block diagram in Fig. 2 are assumed unknown a priori, and must be determined based on the incoming signal. The quality of each of the estimators and classifiers of the autonomous radio is limited by its lack of knowledge of any of the other parameters. As such, the order in which the estimations/classifications are performed is critical. For example, it would not be feasible to classify the modulation type prior to classifying the data rate and obtaining the symbol timing. Using conventional estimation and tracking designs, one quickly gets into a chicken and egg problem, with nearly every estimator needing the output of the other estimators before it can make a maximum likelihood estimate.

To resolve this problem, we have arranged the estimators/classifiers in the partially ordered set shown in Fig. 3, which defines the order in which they may be operated, at least sub-optimally, during the first iteration of estimation. In this ordering, estimators at one level are performed before estimators at the lower levels. The estimators are arranged into the minimum number of levels for which some level i estimators must be performed before level $i+1$ estimators. This approach yields a workable boot-strapping approach to estimating/classifying all of the parameters necessary for the proper operation of the entire receiver.

IV. Estimators and Classifiers of the Autonomous Radio

After identifying the proper order for the estimator modules shown in Fig. 3, we embarked on a design effort for the individual modules. At first glance, it may seem that some of these estimator modules are simply long-established, conventional designs. For example, phase tracking loops have been designed and analyzed for nearly every reasonable signal type. However, we were unable to find any literature for the design of a phase tracking loop for suppressed carrier signals in which the modulation type is unknown. We need a loop that works adequately for any phase modulated signal, and which can improve its performance by later taking input from the modulation classifier when it starts producing an output.

The other seemingly-standard modules had similar design challenges because of unknown signal attributes. Conventional implementations of frequency estimators, symbol timing loops,

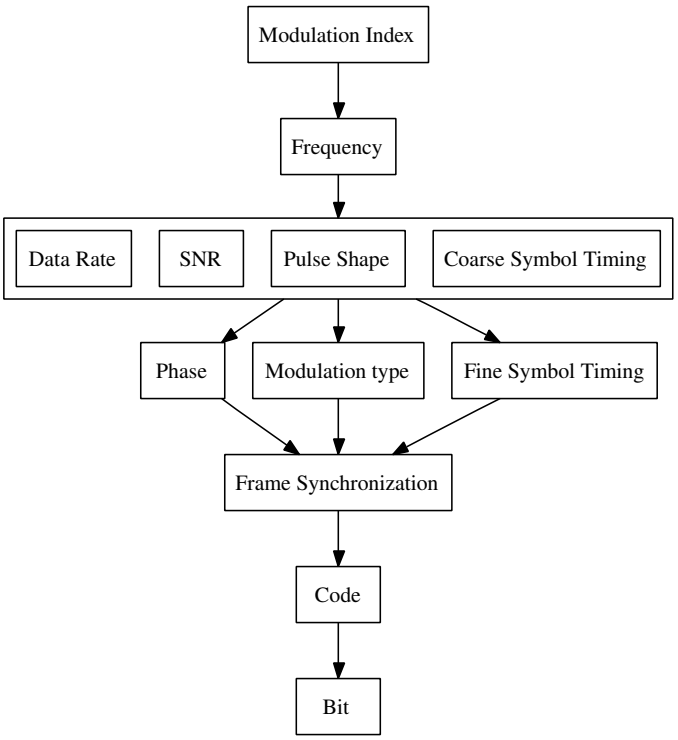

Fig. 3. Estimator dependency graph.

and SNR estimators also assume the modulation type is known.

In addition, there are a number of estimators that are not conventional, and only occur in an autonomous radio. These include the blocks that estimate or classify the data rate, modulation index, modulation type, and code type. We have designed each of these from scratch, in most cases by formulating the maximum likelihood criterion for the estimator and attempting to solve it analytically. This lead to excellent solutions for modulation classification, SNR estimation, and frequency tracking. In some other cases, the maximum likelihood solution was not tractible, and promising ad hoc schemes were identified.

Because the design effort for the individual modules can be quite involved, we are unable to fully describe the design and operation of each module here. Instead, we offer a brief summary of the modules below. The overall structure of the radio is shown in Fig. 4, which shows which parameter estimations are used by each subsequent estimator module.

A. Modulation index estimation

In the initial, or *coarse*, estimation phase, the modulation index estimator requires no parameter estimates, other than the minimum symbol period allowed by the system. It directly estimates the carrier and data powers by integrating over a sufficient epoch. In the subsequent, *fine* estimation phase(s), it may operate coherently and use knowledge of the date rate, modulation type, symbol boundaries, and pulse shape to improve the quality of its data and carrier power estimates.

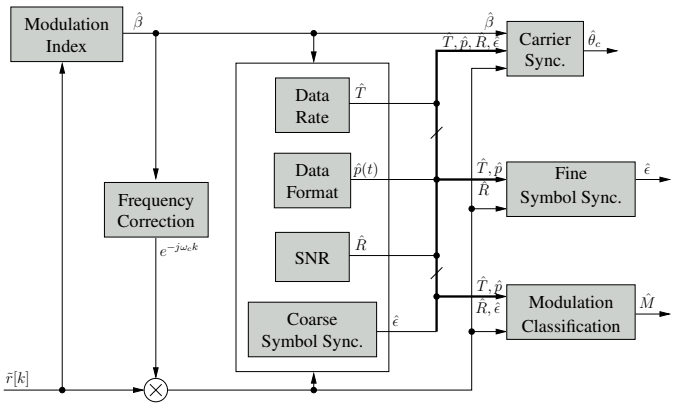

Fig. 4. Signal flow in the autonomous radio.

B. Residual frequency tracking

Frequency correction is performed in the coarse estimation phase only if the modulation index estimator determines that the signal contains a residual carrier ($\hat{\theta}_c < \pi/2$). In that case, an attempt is made to determine the residual carrier frequency via a tracking loop that locks onto the maximum value of a fast Fourier transform (FFT) of the received signal. Several low-complexity designs have been proposed [1] that operate robustly not only over an AWGN channel but also over a Rayleigh fading channel. The structures have been shown to be optimal, in the sense that the optimum open loop estimation of the frequency was derived and found to be the maximum of the FFT, and the closed loop error signal is based on the derivative of this quantity.

C. Joint Estimation of Data Rate, Data Format, SNR, and Coarse Symbol Timing

The joint estimator of the data rate, data format, SNR, and coarse symbol timing takes the frequency-corrected version of the received signal as input, and produces estimates of the symbol period, pulse shape, SNR, and symbol timing offset as output. It operates in the same way during both coarse and fine estimation phases. This is a consequence of the fact the the split-symbol moments estimator (SSME) method of SNR estimation [1] is independent of both the M-PSK modulation order and the carrier phase θ_c.

The basic operation of the module is to compute an SNR estimate under various combinations of hypothesized data rates, data formats, and coarse symbol timings. When the hypothesis is incorrect, the SNR estimate will be low. When the hypothesis is correct, the SNR estimate is more closely related to the true SNR. The hypothesis with the highest corresponding SNR estimate is selected, and those parameters and SNR estimate are output.

The underlying SSME SNR estimator forms its SNR estimation statistic from the sum and difference of information extracted from the first and second halves of each received data symbol. Our initial investigations focused on demonstrating that the scheme, which was previously investigated only for

BPSK modulations, is readily applicable to the class of M-PSK ($M \geq 2$) modulations and furthermore showed that its performance is independent of the value of M. Even more generally, it was pointed out that the complex symbol version of the SSME structure could also be used to provide SNR estimation for two-dimensional signal sets such as quadrature amplitude modulation (QAM). Performance results were obtained for a variety of different scenarios related to the degree of knowledge assumed for the carrier frequency uncertainty and to what extent it is compensated for in obtaining the SNR estimate.

D. Modulation classification

We derived new approximations to the maximum likelihood (ML) classifier to discriminate between M-ary and M'-ary phase-shift-keying (PSK) transmitted on an AWGN channel and received noncoherently, partially coherently, or coherently, and when symbol timing is either known or unknown. A suboptimum classifier can be shown to be ten times less complex than the ML classifier and has less than 0.1 dB performance loss for symbol signal-to-noise ratios (SNR's) in the range (-10,10) dB and any number of observed symbols [1]. Other methods reduce complexity by a factor of 100 with less than 0.2 dB of performance loss. We also developed a classifier that does not require an estimate of the symbol SNR, and a new threshold optimization technique that improves the high-SNR performance of a previously published classifier. We showed that a classification error floor that exists for any classifier on any memoryless channel, even a noiseless one, by deriving a lower bound on the misclassification probability as a function of the number of observed samples.

E. Carrier phase tracking

The carrier synchronizer takes several forms, depending on the modulation index and coarse/fine operation. If the modulation index estimator has determined that there is a residual carrier, then a PLL may be used to lock onto the residual carrier signal. The residual carrier itself is the output of a low pass filter of the received signal, which suppresses the data modulation (except for the portion of the spectrum at DC, if any). When a residual carrier is present, the PLL may be the best choice for carrier synchronization in the fine estimation mode, as well, but this depends on the SNR and the value of modulation index. In cases when the residual carrier is weak, a hybrid loop may outperform the PLL alone in the fine estimation phase.

If the modulation index estimator has determined that the carrier is suppressed, then the carrier synchronization in the coarse estimation phase relies on a carrier loop with passive arms, so that the symbol timing and pulse shape need not be known. A universal loop can be constructed that will work for all M-PSK modulation orders up to some maximum $M_{\max}$ [1]. Since modulation classification is not yet available to the carrier loop during the coarse phase, the carrier loop begins in the coarse estimation phase configured for $M_{\max}$-PSK.

For suppressed carrier signals in the fine estimation phase, the loop can be reconfigured for $\hat{M}$-PSK and the passive arm filters can be replaced with matched filters that make use of the pulse shape and symbol timing estimates, which results in improved performance.

V. An iterative message passing architecture

As mentioned above, the autonomous radio begins by producing estimates at the highest level in Fig. 3 and then proceeding to progressively deeper levels. Initially, no estimator at an upper level can make use of any signal attribute estimated at a level beneath it. This limitation significantly impacts performance, and is inherent to any non-iterative autonomous signal parameter estimation algorithm.

A fundamental innovation of the autonomous radio envisioned here is that after each estimator completes its first estimate in the proper boot-strap order, the deeper level estimators send soft information to the upper estimators. A second iteration then begins, wherein each estimator makes use of this additional extrinsic information to improve its performance. After several iterations, the message passing system will reach a reasonable convergence. We have shown that such coupled systems are typically quite robust, and can provide near maximum likelihood joint estimation/decoding [15], [16], [17].

We now informally describe a nonexhaustive list of the type of soft information that can passed upward during the estimation iterations.

Messages from the symbol timing estimator. The symbol timing module estimates the boundaries of symbol epochs, and can produce a signal that indicates whether it is in lock. The lock indicator, which may be a soft value, can be fed up to the data rate classifier. For example, if the symbol timing tracker is unable to lock onto symbol timing at one data rate, the data rate classifier can make use of that knowledge in reclassifying the data rate.

Messages from the frequency tracker. If the frequency tracker identifies any residual frequency, this may be used to wipe it from the received signal. The SNR estimator performance is affected by residual frequency, and thus, it will be improved by this feedback from the frequency tracking loop.

Messages from the phase tracker. The phase tracking loop output can be used to generate a coherent reference that can be used to improve the symbol timing and SNR estimators, effectively improving the noncoherent performance to coherent performance.

Messages from the modulation classification. Estimates from the modulation classifier can assist in improving SNR, data rate, and symbol timing estimators. The likelihood functions for each modulation have expected values that obey a known relationship to the SNR and symbol timing– for example, the modulation classification becomes more certain with increasing SNR and number of symbol observations. If the observed modulation-type likelihoods are inconsistent with the estimates from the SNR and symbol timing modules, the

likelihoods can be fed back to those modules so they can revise their estimates.

Messages from the decoder. The output of the decoder includes likelihoods for each message bit. Depending on the code, it is usually simple to hard limit these likelihoods and test if the result is a codeword. Typically, codes are designed so that the undetected probability of codeword error is 10^{-10} or lower, which implies that if the output is a codeword, it is nearly certainly the correct one, and no further iterations of the radio are necessary.

If the correct codeword is not obtained, then the bit likelihoods can be used to generate a soft data-wipe of the receieved signal. This makes the signal more like a CW (continuous wave) signal, which will allow the SNR, frequency, and phase estimates to be substantially improved, which will in turn produce better inputs for the decoder to operate on in its second iteration.

VI. A DEMONSTRATION TESTBED

We have developed a software demonstration testbed of the autonomous radio. The testbed contains several parts, as shown in Fig. 5. In one part, the attributes of the transmitted signal may be configured, including the data rate, pulse shape, data format, modulation type, code type, and so forth. Based on these settings, a simulated signal is generated. This is shown in the upper left portion of Fig. 5. The channel parameters, such as carrier frequency offset due to Doppler, timing delays, SNR, and carrier phase are configurable as well, as shown in the lower left portion of Fig. 5.

The data flow of the radio is shown at the top of the figure. Corresponding to each module is a tab in the lower right portion in which the performance of the module may be viewed. E.g., for the modulation classification module, the pre- and post-frequency-corrected complex signal constellations may be viewed. A "System Parameters" tab summarizes all of the true, transmitted parameters, and the corresponding estimations of those parameters that the radio autonomously made.

In the example shown, a 128 kbps suppressed carrier BPSK signal with NRZ pulse shape is transmitted, the symbol SNR is 0 dB, a carrier offset of 15 kHz is applied, the symbol delay is 0.04 symbols, and the carrier phase is 0.4 radians. The autonomous radio corectly identified the signal as suppressed carrier (as seen in the lower right, with a green dot to indicate a match), correctly identified the modulation as BPSK, identified the SNR as -0.27 dB, correctly identified the data rate as 128 kbps, identified a residual frequency of 15.0001 kHz, identified a carrier phase of 3.6 radians, and correctly identified the symbol timing offset of 0.04 symbols. Good data would result from this autonomous configuration of the receiver.

VII. CONCLUSIONS

We have presented an overview of the architecture of an autonomous radio that can receive a signal without much a priori knowledge about its defining characteristics. More details of the designs and analysis of the estimators are available in [1].

Acknowledgments. The author wishes to thank Kenneth Andrews, Michael Cheng, Dariush Divsalar, Sam Dolinar, Fabrizio Pollara, Edgar Satorius, and Biren Shah, Marvin Simon, Andre Tkacenko, Victor Vilnrotter, Mandy Wang, and Hua Xie, for their many contributions and helpful discussions about autonomous radios.

REFERENCES

[1] J. Hamkins and M. Simon, eds., *Autonomous Software-Defined Radio Receivers for Deep Space Applications*. New York: John Wiley & Sons, 2006.

[2] C. D. Edwards, A. Barbieri, E. Brower, P. Estabrook, R. Gibbs, R. Horttor, J. Ludwinski, R. Mase, C. McCarthy, R. Schmidt, P. Theisinger, T. Thorpe, and B. Waggoner, "A Martian telecommunications network: UHF relay support of the Mars Exploration Rovers by the Mars Global Surveyor, Mars Odyssey, and Mars Express Orbiters, IAC-04-M.5.07," in *International Astronautical Congress 2004*, pp. 1–11, Oct. 2004.

[3] J. Guinn, "Phoenix plans for relay use," in *Mars Network Workshop*, June 2004.

[4] R. Barry, "MSL plans for relay use," in *Mars Network Workshop*, June 2004.

[5] J. Oberg, "Titan calling," *IEEE Spectrum*, vol. 41, no. 10, pp. 28–33, 2004.

[6] "Telemetry channel coding. CCSDS 101.0-B-6. Blue Book. Issue 6." Oct. 2002.

[7] "Proximity-1 space link protocol. CCSDS 211.0-B-1. Blue Book. Issue 1." Oct. 2002.

[8] S. Lin and D. J. Costello Jr., *Error Control Coding: Fundamentals and Applications*. New Jersey: Prentice-Hall, 1983.

[9] "CCSDS 201.0-B-3. Telecommand Part 1–Channel Service. Blue Book. Issue 3." June 2000.

[10] J. G. Proakis, *Digital Communications*. New York, NY: McGraw Hill, Inc., third ed., 1995.

[11] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[12] S. W. Asmar, "Characteristic trends of ultrastable oscillators for radio science experiments," *TDA Progress Report*, vol. 42, pp. 1–5, May 1997.

[13] S. Hinedi, "A functional description of the advanced receiver," *TDA Progress Report*, vol. 42, pp. 131–149, Feb. 1990.

[14] C. D. Edwards, T. C. Jedrey, E. Schwartzbaum, A. S. Devereaux, R. DePaula, M. Dapore, and T. W. Fischer, "The Electra proximity link payload for Mars relay telecommunications and navigation, IAC-03-Q.3.A06," in *International Astronautical Congress 2003*, Sep.-Oct. 2003.

[15] D. Divsalar and F. Pollara, "Turbo trellis coded modulation with iterative decoding for mobile satellite communications," in *IMSC*, 1997.

[16] J. Hamkins and D. Divsalar, "Coupled receiver-decoders for low rate turbo codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, June 2003.

[17] J. Gunn, K. Barron, and W. Ruczczyk, "A low-power DSP core-based software radio architecture," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 574–590, Apr. 1999.

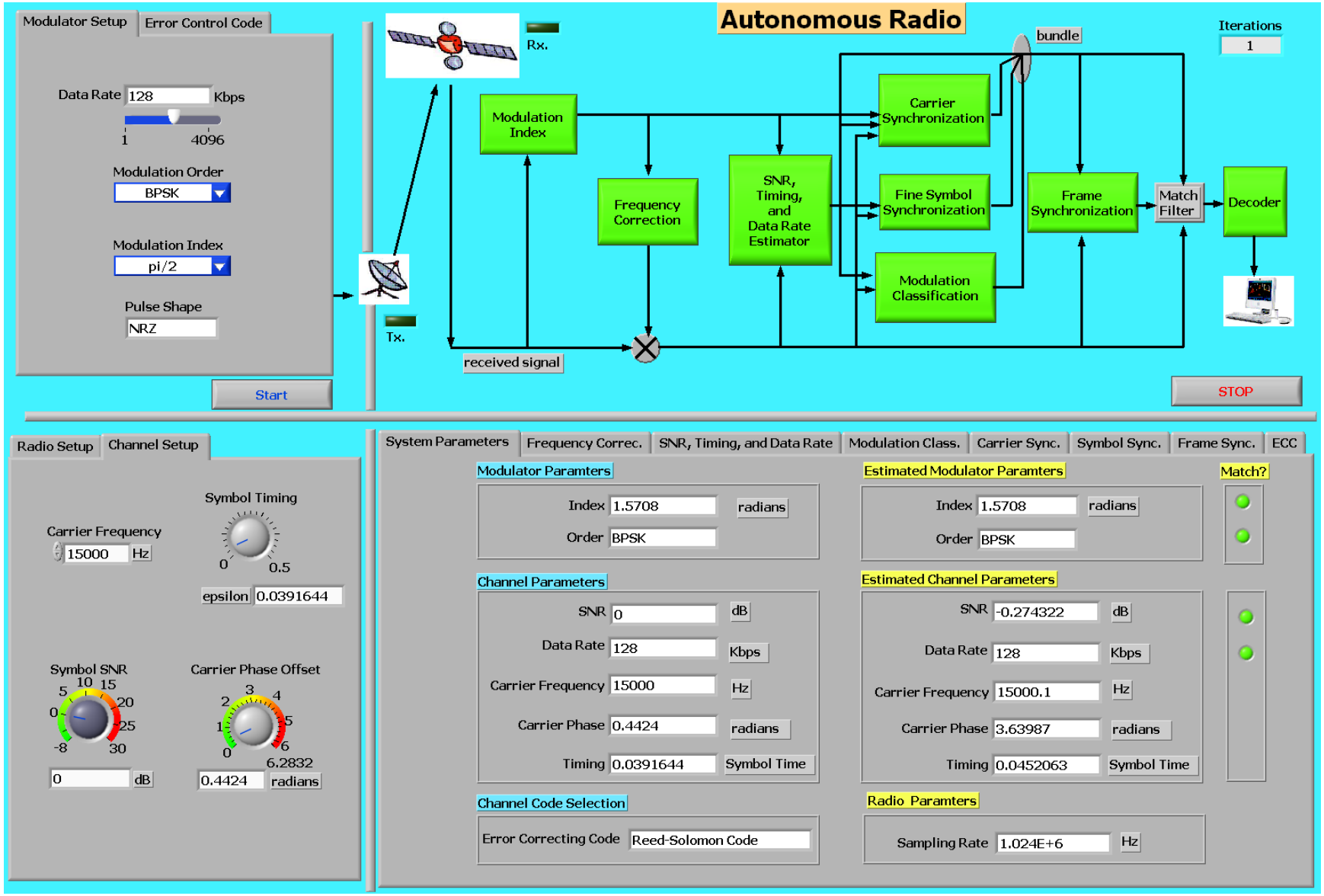

Fig. 5. Screenshot of the autonomous radio software testbed.